



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,583	12/31/2003	Hong Jiang	ITL.1704US (P17510)	8582
21906 7590 03/13/2009 TROP, PRUNER & HU, P.C. 1616 S. VOSS ROAD, SUITE 750 HOUSTON, TX 77057-2631			EXAMINER WAI, ERIC CHARLES	
			ART UNIT 2195	PAPER NUMBER
			MAIL DATE 03/13/2009	DELIVERY MODE PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.



### DETAILED ACTION

1. Claims 1, 3, 6-7, and 11-21 are presented for examination.

#### ***Double Patenting***

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 1, 3, 6-7, and 11-21 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-5, 12-13, 19-22, 26-27, and 33-34 of copending Application No. 10/750,589. Although the conflicting claims are not identical, they are not patentably distinct from each other.
4. For example, claim 1 of copending Application No. 10/750,589 recites placing a thread in an inactive state in response to a predetermined condition and sending a

Art Unit: 2195

message from a semaphore to change the state of the thread. Claim 11 of the present application performs the substantially the same steps. Claim 1 of copending Application No. 10/750,589 differs only in that the threads are intended to be used to process graphical elements of an image. It would have been obvious to one of ordinary skill to try to extend the teachings to image processing.

5. This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

### ***Claim Rejections - 35 USC § 102***

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1, 3, and 6 are rejected under 35 U.S.C. 102(b) as being anticipated by Wang et al. (US Pat No. 5,771,382).

8. Regarding claim 1, Wang teaches a method comprising:

placing an executable thread of instructions in an inactive state in response to a resource being unavailable (col 6 lines 61-63, wherein threads that do not have access to the resource are directed to a wait state; col 8 lines 1-19, wherein an embodiment of the wait state includes putting the thread to sleep); and

Art Unit: 2195

when the resource becomes available, changing the thread of instructions to the active state and granting the resource to the thread of instructions (col 6 lines 63-65).

9. Regarding claim 3, Wang teaches executing the thread of instructions when in the active state (col 8 lines 60-62).

10. Regarding claim 6, Wang teaches maintaining an indication of a state of each of a plurality of executable threads of instructions (col 6 lines 61-65, wherein it is inherent that Wang maintains an indication on the state of each thread in order to change them from the wait state).

### ***Claim Rejections - 35 USC § 103***

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

12. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Wang et al. (US Pat No. 5,771,382).

Art Unit: 2195

13. Regarding claim 7, Wang does not teach that the indication of the state of each thread comprises a state variable corresponding to a dependency, if any, of an associated thread.

14. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify Wang to include a state variable corresponding to a dependency of an associated thread. Wang teaches that each thread is dependent on the use of the semaphore/resource for operation. Wang also teaches the need to synchronize communications among threads (col 2 lines 33-34). One would be motivated by the desire to prohibit out of order accesses to resources that could cause system conflicts.

15. Claims 11-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wenniger (US Pat No. 6,018,785).

16. Regarding claim 11, Wenniger teaches an apparatus comprising:

an execution circuit to receive and execute a thread of instructions, wherein the execution circuit transmits a semaphore request message and places the thread in an inactive state in response to the thread of instructions requiring a resource having an associated semaphore (col 6 lines 12-22, wherein the requesting process must await an interrupt from the semaphore); and

a semaphore entity coupled with the execution circuit to receive the semaphore request message from the execution circuit and to selectively grant control of the

Art Unit: 2195

semaphore in response to the semaphore request message by transmitting a semaphore acknowledge message to the execution circuitry, wherein the execution circuitry, in response to receiving the semaphore acknowledge message, removes the thread of instructions from the inactive state and grants the resource to the thread when the resource becomes available (col 6 lines 12-22, where upon receiving the interrupt, the thread queries the semaphore; col 6 lines 40-42, wherein the resource is granted).

17. Wenniger does not explicitly teach that the thread of instructions is paced in an inactive state. Wenniger only teaches that the thread awaits the interrupt from the hardware semaphore. However, it would have been obvious to one of ordinary skill in the art at the time of the invention, that the thread would be placed in an inactive state. It is well known in the art that threads are stalled when the resources that they require are unavailable. One would be motivated by the desire to reduce idle execution time of threads awaiting resources as is well known in the art.

18. Regarding claim 12, Wenniger does not teach further comprising: at least one additional execution circuit to execute threads of instructions; and a thread dispatcher coupled with the execution circuit and at least one additional execution circuit to dispatch threads for execution by selected execution circuits.

19. It would have been obvious to one of ordinary skill in the art, at the time of the invention to add one additional execution circuit to execute threads of instructions and a thread dispatcher. It is well known in the art to add additional execution units to increase processing capability of processors.

20. Regarding claim 13, Wenniger teaches that the execution circuitry, in response to receiving the semaphore acknowledge message, resumes execution of the thread of instructions including accessing the resource associated with the semaphore (col 6 lines 12-22).

21. Regarding claim 14, Wenniger teaches that when the thread of instructions is in the inactive state, execution of the instructions ceases and the execution circuitry does not poll the semaphore entity to determine a status of the semaphore request message (col 6 lines 6-12).

22. Regarding claims 15-18, they are the system claims of claims 11-14 above. Therefore, they are rejected for the same reasons as claims 11-14 above.

23. Claims 19-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang et al. (US Pat No. 5,771,382) in view of Winkeler et al. (US Pat No. 7,237,013).

24. Regarding claim 19, Wang does not teach placing requests for a semaphore in a queue.

25. However, Winkeler teaches a well known technique of creating a semaphore queue to queue pending requests (col 10 lines 39-47). It would have been obvious to



Art Unit: 2195

one of ordinary skill in the art at the time of the invention to modify Wang to teach placing requests for a semaphore in a queue. One would be motivated by the desire to keep track of processes that desire access of the resource by queuing them.

26. Regarding claim 20, Wang teaches causing a thread to release a semaphore when use of a resource is completed (col 6 lines 63-65).

27. Regarding claim 21, Winkeler teaches automatically granting the resource to the thread whose request is the next request in the queue (col 10 lines 45-47, wherein each component can obtain the lock "in turn").

### ***Response to Arguments***

28. Applicant's arguments filed 12/10/2008 have been fully considered but they are not persuasive.

29. Applicant argues on pg 5 of Remarks regarding claim 1:

"As explained in column 3, lines 28-35, "the present invention provides a means to eliminate this problem" where "this problem" is that in a multithreaded application, a static variable can be initialized more than once. Thus, there is no need to grant the variable to a thread after it has been initialized. It is not a resource that can only be used by one thread at a time. It is simply a variable that should not be used by any thread

Art Unit: 2195

until any one thread initializes it. Thus, once it has been initialized, all the threads can use it. Therefore, there is no teaching, nor any reason, to grant the resource to the thread of instructions after changing the thread to the active state.”

30. Examiner disagrees. Applicant asserts that variables are not the same as resources. It is well known in the art that variables are known to a form of computer resources. Microsoft Computer Dictionary (5<sup>th</sup> Edition, 2002) describes a resource as “any nonexecuting data that is logically deployed with an application ... resources can contain data in a number of forms, including strings, images, and persisted objects”. Furthermore, while Wang indicates that solutions regarding the initialization of variables, Wang’s disclosure further reinforces that variables are resources (col 6 lines 7-23, wherein “control of the semaphore indicates which thread has exclusive control of *machine resources*”).

31. Applicant’s argument that there is no need to grant the variable to a thread after it has been initialized is inapposite. Claim 1 simply requires that “when the resources become available, changing the thread of instructions to the active state and granting resources to the thread of instructions”. The claim language does not require that granting be continuously performed. It only requires that granting to be performed at some point in time. Wang clearly teaches the granting the threads access to the variable after it has been unlocked. The issue of allowing one or all of the threads access is irrelevant since the claim language only requires that access to the resource be granted.

Art Unit: 2195

32. Applicant argues on pg 5 of Remarks regarding claim 1:

“Likewise, claim 15 calls for an execution unit to grant the resource to the thread of instructions when the resource becomes available. Here, in the cited reference, the resource never really becomes available and it is never granted to any thread after it is initialized. The threads are just prevented from using it until such time as it is initialized and then there is nothing to stop them all from using it.”

33. Examiner disagrees. Wang teaches claim 15 for the same reasons argued above. Wang clearly teaches that threads without control of the semaphore must wait until control of the semaphore is relinquished (col 6 lines 21-23). In Wang, the threads all seek access to the resource and are directed to sleep after the first thread performs the locking step (col 7 lines 61-67). It is clear that access to the resources is denied to all threads except for the first thread via the locking step (col 6 lines 61-63). The threads are then allowed to continue execution (i.e. allowed to access the variables) when the first thread performs an unlocking step (col 8 lines 60-62). Applicant admits that the threads are ‘prevented’ from using the resource. Therefore, a granting step occurs in order for the threads to access the resource.

### ***Conclusion***

34. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

Art Unit: 2195

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

35. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Eric C. Wai whose telephone number is 571-270-1012. The examiner can normally be reached on Mon-Thurs, 9am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng - Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2195

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Meng-Ai An/  
Supervisory Patent Examiner, Art Unit 2195

/Eric C Wai/  
Examiner, Art Unit 2195